Question 1

Which of the following are absolutely necessary for creating a functioning shiny app? (Check all that apply)

Answer: A server.R file containing a call to shinyServer() A ui.R file containing a call to shinyUI()

Question 2

What is incorrect about the following syntax in ui.R?

**library**(shiny)

shinyUI(pageWithSidebar(

headerPanel("Data science FTW!"),

sidebarPanel(

h2('Big text')

h3('Sidebar')

),

mainPanel(

h3('Main Panel text')

)

))

Answer: Missing a comma in the sidebar panel

Question 3

Consider the following in ui.R

shinyUI(pageWithSidebar(

headerPanel("Example plot"),

sidebarPanel(

sliderInput('mu', 'Guess at the mu',value = 70, min = 60, max = 80, step = 0.05,) ),

mainPanel(

plotOutput('newHist')

)

))

And the following in server.R

**library**(UsingR)

data(galton)

shinyServer(

**function**(input, output) {

output$myHist <- renderPlot({

hist(galton$child, xlab='child height', col='lightblue',main='Histogram')

mu <- input$mu

lines(c(mu, mu), c(0, 200),col="red",lwd=5)

mse <- mean((galton$child - mu)^2)

text(63, 150, paste("mu = ", mu))

text(63, 140, paste("MSE = ", round(mse, 2)))

}) }

)

Answer: The server.R output name isn’t the same as the plotOutput command used in ui.R.

Question 4

What are the main differences between creating a Shiny Gadget and creating a regular Shiny App? (Check all that apply)

Answer Shiny Gadgets are designed to have small user interfaces that fit on one page. Shiny Gadgets are designed to be used by R users in the middle of a data analysis.

Question 5

Consider the following R script:

**library**(shiny)

**library**(miniUI)

pickXY <- **function**() {

ui <- miniPage(

gadgetTitleBar("Select Points by Dragging your Mouse"),

miniContentPanel(

plotOutput("plot", height = "100%", brush = "brush")

)

)

server <- **function**(input, output, session) {

output$plot <- renderPlot({

plot(data\_frame$X, data\_frame$Y, main = "Plot of Y versus X",

xlab = "X", ylab = "Y")

})

observeEvent(input$done, {

stopApp(brushedPoints(data\_frame, input$brush,

xvar = "X", yvar = "Y"))

})

}

runGadget(ui, server)

}

my\_data <- data.frame(X = rnorm(100), Y = rnorm(100))

pickXY(my\_data)

Answer: No arguments are defined for pickXY()